Task 4: EN-TEx ATAC-seq data: downstream analyses

Before starting with the exercise, I enter ENCODE portal to download sigmoid and stomach data for ATA-seq analysis.
These the code for each tissue:

ENCFF287UHP	UBERON:0001159	sigmoid colon
ENCFF762IFP	UBERON:0000945	stomach

Let”s start with the analysis

1) Enter the docker container:

docker run -v $PWD:$PWD -w $PWD --rm -it dgarrimar/epigenomics_course

2) Move to the ATAC-sec folder:

cd epigenomics_uvic/ATAC-seq/

3) Create folders to store bigBed data files and peaks analysis files, as performed in class with Chip-seq analysis:

mkdir data
mkdir annotation
mkdir analyses

mkdir data/bigBed.files
mkdir data/bed.files
mkdir analysese/peaks

4) Retrieve from a newly generated metadata file ATAC-seq peaks (bigBed narrow, pseudoreplicated peaks, assembly GRCh38) for stomach and sigmoid_colon for the same donor used in the previous sections.

I downloaded metadata using the URL code present in the txt files downloaded form ENCODE portal

../bin/download.metadata.sh "https://www.encodeproject.org/metadata/?replicates.library.biosample.donor.uuid=d370683e-81e7-473f-8475-7716d027849b&status=released&status=submitted&status=in+progress&biosample_ontology.term_name=sigmoid+colon&biosample_ontology.term_name=stomach&assay_title=ATAC-seq&assay_slims=DNA+accessibility&type=Experiment"

5) Check the download metadata:

head -1 metadata.tsv | awk 'BEGIN{FS=OFS="\t"}{for (i=1;i<=NF;i++){print $i, i}}'

6) From metadata file, greped for subsetting for bigBed_narrow, pseudoreplicated_peaks and GRCh38 and download the data that matches the subseted parameters on bigBed.peaks.ids.txt

grep -F "bigBed_narrowPeak" metadata.tsv |
grep -F "pseudoreplicated_peaks" |
grep -F "GRCh38" |

```
awk 'BEGIN{FS=OFS="\t"}{print $1, $10, $22}' |\
sort -k2,2 -k1,1r |\
sort -k2,2 -u > analyses/bigBed.peaks.ids.txt

cut -f1 analyses/bigBed.peaks.ids.txt |\
while read filename; do
  wget -P data/bigBed.files "https://www.encodeproject.org/files/$filename/@@download/
$filename.bigBed"
done
```

7) For each tissue, run an intersection analysis using BEDTools:
First of all convert bigBed files of H3K4me3 peaks to BED files

```
mkdir data/bed.files/
cut -f1 analyses/bigBed.peaks.ids.txt |\
while read filename; do
  bigBedToBed data/bigBed.files/"$filename".bigBed data/bed.files/"$filename".bed
done
```

Then retrieve genes with peaks of H3K4me3 at the promoter region in each tissue.

```
cut -f-2 analyses/bigBed.peaks.ids.txt |\
while read filename tissue; do
  bedtools intersect -a annotation/gencode.v24.protein.coding.non.redundant.TSS.bed -b data/
bed.files/"$filename".bed -u |\
  cut -f7 |\
  sort -u > analyses/peaks.analysis/genes.with.peaks."$tissue".H3K4me3.txt
done

cut -f-2 analyses/bigBed.peaks.ids.txt |\
while read filename tissue; do
  bedtools intersect -a annotation/gencode.v24.protein.coding.non.redundant.TSS.bed -b data/
bed.files/"$filename".bed -u |\
  cut -f7 |\
  sort -u > analyses/peaks.analysis/genes.with.ATAC.peaks."$tissue".txt
done
```

Report the number of peaks that intersect promoter regions

```
wc -l analyses/peaks.analysis/genes.with.ATAC.peaks.UBERON\:0000945.txt
```
15029 peaks
```
wc -l analyses/peaks.analysis/genes.with.ATAC.peaks.UBERON\:0001159.txt
```
14830 peaks

Report the number of peaks that fall outside gene coordinates (whole gene body, not just the promoter regions)

```
cut -f-2 analyses/bigBed.peaks.ids.txt |\
while read filename tissue; do
  bedtools intersect -a data/bed.files/"$filename".bed -b  annotation/
gencode.v24.protein.coding.gene.body.bed -v > data/bed.files/
ATAC.peaks.outside.gene.body."$tissue".bed
```

done

wc -l data/bed.files/ATAC.peaks.outside.gene.body.UBERON\:0000945.bed
34537 peaks
wc -l data/bed.files/ATAC.peaks.outside.gene.body.UBERON\:0001159.bed
37035 peaks

Task 5. Distal regulatory activity

1) Create a folder regulatory_elements inside epigenomics_uvic.

cd ..
mkdir regulatory_elements
mkdir regulatory_elements/analyses
mkdir regulatory_elements/data
mkdir regulatory_elements/annotation
mkdir regulatory_elements/analyses/peaks.analysis/
mkdir regulatory_elements/data/bed.files
mkdir regulatory_elements/data/bigBed.files

cut -f-2 ATAC-seq/analyses/bigBed.peaks.ids.txt |
while read filename tissue; do
cp ATAC-seq/data/bed.files/ATAC.peaks.outside.gene.body."$tissue".bed regulatory_elements/data/bed.files/
done

2) Distal regulatory regions are usually found to be flanked by both H3K27ac and H3K4me1. From your starting catalogue of open regions in each tissue, select those that overlap peaks of H3K27ac AND H3K4me1 in the corresponding tissue. You will get a list of candidate distal regulatory elements for each tissue. How many are they?

cd regulatory_elements
../bin/download.metadata.sh "https://www.encodeproject.org/metadata/?type=Experiment&replicates.library.biosample.donor.uuid=d370683e-81e7-473f-8475-7716d027849b&status=released&assembly=GRCh38&biosample_ontology.term_name=sig

H3K4me1 peaks:

grep -F H3K4me1 metadata.tsv |
grep -F "bigBed_narrowPeak" |
grep -F "pseudoreplicated_peaks" |
grep -F "GRCh38" |
awk 'BEGIN{FS=OFS="\t"}{print $1, $10, $22}' |
sort -k2,2 -k1,1r |
sort -k2,2 -u > analyses/bigBed.peaks.ids.H3K4me1.txt

cut -f1 analyses/bigBed.peaks.ids.H3K4me1.txt |
while read filename; do

```
  wget -P data/bigBed.files "https://www.encodeproject.org/files/$filename/@@download/
$filename.bigBed"
done
```

Convert bigBed files to BED files

```
cut -f1 analyses/bigBed.peaks.ids.H3K4me1.txt |\
while read filename; do
  bigBedToBed data/bigBed.files/"$filename".bigBed data/
bed.files/"$filename".H3K4me1.bed
done
```

H3K27ac peaks:

```
grep -F H3K27ac metadata.tsv |\
grep -F "bigBed_narrowPeak" |\
grep -F "pseudoreplicated_peaks" |\
grep -F "GRCh38" |\
awk 'BEGIN{FS=OFS="\t"}{print $1, $10, $22}' |\
sort -k2,2 -k1,1r |\
sort -k2,2 -u > analyses/bigBed.peaks.ids.H3K27ac.txt

cut -f1 analyses/bigBed.peaks.ids.H3K27ac.txt |\
while read filename; do
  wget -P data/bigBed.files "https://www.encodeproject.org/files/$filename/@@download/
$filename.bigBed"
done
```

Convert bigBed files to BED files

```
cut -f1 analyses/bigBed.peaks.ids.H3K27ac.txt |\
while read filename; do
  bigBedToBed data/bigBed.files/"$filename".bigBed data/bed.files/"$filename".H3K27ac.bed
done
```

Overlapping of H3K4me1 peaks

```
cut -f-2 analyses/bigBed.peaks.ids.H3K4me1.txt |\
while read filename tissue; do
  bedtools intersect -a data/bed.files/"$filename".H3K4me1.bed -b data/bed.files/
ATAC.peaks.outside.gene.body."$tissue".bed -u |\
  sort -u > data/bed.files/ATAC.H3K4me1.overlapping.peaks.outside.gene.body."$tissue".bed
done
```

Overlapping of H3K27ac peaks

```
cut -f-2 analyses/bigBed.peaks.ids.H3K27ac.txt |\
while read filename tissue; do
  bedtools intersect -a data/bed.files/"$filename".H3K27ac.bed -b data/bed.files/
ATAC.H3K4me1.overlapping.peaks.outside.gene.body."$tissue".bed -u |\
  sort -u > data/bed.files/candidate.distal.regulatory.region."$tissue".bed
done
```

Peaks in stomach

wc -l data/bed.files/candidate.distal.regulatory.region.UBERON\:0000945.bed
4543 peaks

Peaks in sigmoid_colon

wc -l data/bed.files/candidate.distal.regulatory.region.UBERON\:0001159.bed
7853 peaks

3) Focus on regulatory elements that are located on chromosome 1 (hint: to parse a file based on the value of a specific column, have a look at what we did here), and generate a file regulatory.elements.starts.tsv that contains the name of the regulatory region (i.e. the name of the original ATAC-seq peak) and the start (5') coordinate of the region.

```
mkdir data/tsv.files |\
cut -f2 analyses/bigBed.peaks.ids.H3K27ac.txt |\
while read tissue; do
  grep -w "chr1" data/bed.files/candidate.distal.regulatory.region."$tissue".bed | awk
'BEGIN{FS=OFS="\t"}{print $4, $2}' > data/tsv.files/regulatory.elements.starts."$tissue".tsv
done
```

4) Focus on protein-coding genes located on chromosome 1. From the BED file of gene body coordinates that you generated here, prepare a tab-separated file called gene.starts.tsv which will store the name of the gene in the first column, and the start coordinate of the gene on the second column (REMEMBER: for genes located on the minus strand, the start coordinate will be at the 3'). Use the command below as a starting point:

```
cp ../ChIP-seq/annotation/gencode.v24.protein.coding.gene.body.bed annotation/
 grep -w "chr1" annotation/gencode.v24.protein.coding.gene.body.bed | awk
'BEGIN{FS=OFS="\t"}{if ($6=="+"){start=$2} else {start=$3}; print $4, start}' >
annotation/gene.starts.tsv
```

5) Download or copy this python script inside the epigenomics_uvic/bin folder. Have a look at the help page of this script to understand how it works:

```
cd ../bin/
wget https://public-docs.crg.es/rguigo/Data/bborsari/UVIC/epigenomics_course/
get.distance.py
python ../bin/get.distance.py -h
```

```
##############################################################################
Modification of python script
nano ..bin/get.distance.py

#********
# BEGIN *
#********

x=1000000 # set maximum distance to 1 Mb
selectedGene="" # initialize the gene as empty
```

```
selectedGeneStart=0 # initialize the start coordinate of the gene as empty

for line in open_input.readlines(): # for each line in the input file
        gene, geneStart = line.strip().split('\t') # split the line into two c$
        position = int(geneStart) # define a variable called position that cor$
        difference = abs(position - enhancer_start) # compute the absolute val$

        if difference < x: # if this absolute value is lower than x
                x = difference # this value will now be your current x
                selectedGene = gene # save gene as selectedGene
                selectedGeneStart = position # save position as selectedGeneSt$

print "\t".join([selectedGene, str(selectedGeneStart)
, str(x)])
```

##############3###

cd ../regulatory_elements
python ../bin/get.distance.py --input annotation/gene.starts.tsv --start 980000

ENSG00000187642.9 982093 2093

6) For each regulatory element contained in the file regulatory.elements.starts.tsv, retrieve the closest gene and the distance to the closest gene using the python script you created above.

For stomach:
cat data/tsv.files/regulatory.elements.starts.UBERON\:0000945.tsv | while read element start; do python ../bin/get.distance.py --input annotation/gene.starts.tsv --start "$start"; done > analyses/regulatoryElements.genes.distances.0000945.tsv

For sigmoid_colon
cat data/tsv.files/regulatory.elements.starts.UBERON\:0001159.tsv | while read element start; do python ../bin/get.distance.py --input annotation/gene.starts.tsv --start "$start"; done > analyses/regulatoryElements.genes.distances.0001159.tsv

7) Use R to compute the mean and the median of the distances stored in regulatoryElements.genes.distances.tsv.

```
#############################################################################
R script to calculate mean and median
setwd("D:/Descargas")
stomach<-read.delim("regulatoryElements.genes.distances.0000945.tsv",header=FALSE)
sigmoid_colon<-read.delim("regulatoryElements.genes.distances.
0001159.tsv",header=FALSE)
mean_stomach<-mean(stomach$V3)
#[1] 47810.49
median_stomach<-median(stomach$V3)
#[1] 27245
mean_sigmoid_colon<-mean(sigmoid_colon$V3)
#[1] 76557.72
median_sigmoid_colon<-median(sigmoid_colon$V3)
#[1] 36643.5
```